Conf-760808--2

LA-UR-76-594

TITLE: COMPUTING THE SMITH NORMAL FORM OF A MATRIX

AUTHOR(S): Jo Ann Howell, C-3

SUBMITTED TO: SYMSAC 76
SIGSAM MEETING, YORKTOWN HEIGHTS
AUGUST 10-12, 1976

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

MASTER

Form No. 836
St. No. 2629
1/75

UNITED STATES
ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

Computing the Smith Normal Form of a Matrix*

Jo Ann Howell

1. Introduction

The reduction of a matrix to a normal form enables us to study the matrix in its simplest and most convenient shape, and to more immediately relate the theory of matrices to scientific applications. We study in this paper an algorithm for computing symbolically the Smith normal form of a matrix. First, we introduce some basic concepts. Further background is found in Gantmacher [1960, pp.130-174] or Turnbull and Aitken [1961, pp.21-28].

Let $A(\lambda)$ be an m×n matrix having polynomial elements with coefficients over a field F. We can write

$$A(\lambda) = A_k \lambda^k + A_{k-1} \lambda^{k-1} + \ldots + A_0,$$

where the A_i are m×n matrices with elements over F. $A(\lambda)$ is called a λ-matrix.

Every λ-matrix of rank r can be reduced by elementary transformations (rational in the field of elements of $A(\lambda)$) to a diagonal form containing exactly r nonzero elements,

$$B(\lambda) = P(\lambda)\, A(\lambda)\, Q(\lambda)$$

$$= \begin{bmatrix} E_1(\lambda) & & & & & \bigcirc \\ & E_2(\lambda) & & & & \\ & & \ddots & & & \\ & \bigcirc & & E_r(\lambda) & & \\ & & & & 0 & \\ & & & & & \ddots \; 0 \end{bmatrix}.$$

*Los Alamos Scientific Laboratory of the University of California, Los Alamos, New Mexico 87544. This research was supported by the U.S. Energy Research and Development Administration.

$P(\lambda)$ and $Q(\lambda)$ are square λ-matrices with nonzero determinants independent of λ. Each $E_i(\lambda)$ is a monic polynomial in λ such that $E_i(\lambda)$ divides $E_{i+1}(\lambda)$. The polynomials $E_i(\lambda)$ are called the invariant factors of $A(\lambda)$. This diagonal form is known as the Smith normal form for equivalent λ-matrices.

2. Algorithm for Computing the Smith Normal Form

Hereafter we shall assume that A is an m×n λ-matrix and shall omit the (λ). ROW and COLUMN are described below. For related algorithms and discussion see Bradley [1971].

```
SMITH:
Step 1:   t ← min(m,n)
Step 2:   [Construct diagonal form row by row.]
          For i = 1 ,..., t-1 do steps 3-8
Step 3:   [Check for a zero row.]
          While row i of A is 0 do
               If i<t-1 then i←i+1
               else go to 9
          end
Step 4:   For j = i+1 ,..., m do
               If remainder (A_{j,i},A_{i,i})≠0 do
                    ROW(A,i)
                    go to Step 5
               end
          end
Step 5:   For j=i+1 ,..., n do
               If remainder (A_{i,j},A_{i,i})≠0 do
                    COLUMN(A,i)
                    go to step 4
               end
          end
```

```
Step 6:   [Subtract multiples of column i from other columns.]
          For j = i+1 ,..., n do
               For k = i ,..., m do
                    A_{k,j} ← A_{k,j} - (A_{i,j}/A_{i,i}) A_{k,i}
               end
          end
Step 7:   For j = i+1 ,..., m      A_{j,i} ← 0
Step 8:   [Make pivotal element monic.]
          A_{i,i} ← A_{i,i}/ldcf(A_{i,i})      (ldcf is leading coefficient)
Step 9:   [Make last pivotal element monic.]
          A_{t,t} ← A_{t,t}/ldcf(A_{t,t})
Step 10:  If m < n then do
               For j = m+1 ,..., n     A_{m,j} ← 0
          end
Step 11:  If n < m then do
               For j = n+1 ,..., m     A_{j,n} ← 0
          end
Step 12:  For i = 1 ,..., t - 1
               For k = i+1 ,..., t
                    If remainder (A_{k,k},A_{i,i}) ≠ 0 then do
                         g ← gcd (A_{k,k},A_{i,i})
                         A_{k,k} ← A_{i,i} A_{k,k}/g
                         A_{i,i} ← g
                    end
               end
          end
```

Using the function ROW we perform elementary column operations on the ith, (i+1)th ,..., nth column of A until $A_{i,i}$

divides $A_{i,j}$, $j = i+1,\dots,n$. Rows i to m of A are affected by the transformations.

ROW:

Step 1: [Make elements in row i monic.]

For $\ell = i, \dots, n$ do

For $j = i, \dots, m$ $A_{j,\ell} \leftarrow A_{j,\ell}/\mathrm{ldcf}(A_{i,\ell})$

end

Step 2: [Find the element of lowest degree in row i.]

Set k to the column number such that

$$\deg(A_{i,k}) \le \deg(A_{i,j}) \ , j=i,\dots,n,$$

and $A_{i,k} \neq 0$.

Step 3: [Interchange columns k and i, if $k \neq i$.]

For $j = i, \dots, m$ Exchange $A_{j,k}$ and $A_{j,i}$

Step 4: Calculate x_j such that

$$\gcd(A_{i,i}, A_{i,i+1}, \dots, A_{i,n}) = x_i A_{i,i} + x_{i+1} A_{i,i+1} + \dots + x_n A_{i,n}$$

Step 5: [Steps 5-8 are special cases.]

For $k = 1, \dots, n$ do

If $x_k = 1$ or $A_{i,k} = 0$ then go to step 12

end

Step 6: For $k = 1, \dots, n$ do

If $x_k = -1$ then do

For $j = i, \dots, m$ $A_{j,k} \leftarrow -A_{j,k}$

go to step 12

end

end

Step 7: For $k = i+1, \ldots, n$ do

  If $A_{i,i}$ divides $A_{i,k}$ then do

    $x_i \leftarrow x_i - A_{i,k} / A_{i,i}$

    go to step 12

  end

end

Step 8: For $k = i, \ldots, n$ do

  If $x_k = 0$ then do

    $d \leftarrow A_{i,k} / \gcd(A_{i,i}, \ldots, A_{i,n})$

    For $j = i, \ldots, n$ $x_j \leftarrow (1-d) x_j$

    go to step 12

  end

end

Step 9: Calculate y_1 and y_2 such that

  $g = \gcd(A_{i,i}, A_{i,i+1}) = y_1 A_{i,i} + y_2 A_{i,i+1}$

Step 10: $z_1 \leftarrow -A_{i,i+1}/g$

$z_2 \leftarrow A_{i,i}/g$

Step 11: [Put gcd in $A_{i,i}$ and 0 in $A_{i,i+1}$.]

For $j = i, \ldots, m$ do

  $d \leftarrow y_1 A_{j,i} + y_2 A_{j,i+1}$

  $A_{j,i+1} \leftarrow z_1 A_{j,i} + z_2 A_{j,i+1}$

  $A_{j,i} \leftarrow d$

end

go to step 4

Step 12: [Replace pivotal element with gcd.]

For $j = i, \ldots, n$ $(j \neq k)$ do

  For $\ell = i, \ldots, m$ $A_{\ell,k} \leftarrow A_{\ell,k} + x_j A_{\ell,j}$

end

Step 13: [Interchange columns i and k.]

For $j = i, \ldots, m$ Interchange $A_{j,i}$ and $A_{j,k}$

COLUMN(A,i) is the same as ROW(A^T,i).

A key operation encountered in this reduction is the computation of multipliers $x_1, \ldots, x_n$ such that

$$\sum_{i=1}^{n} a_i x_i = \gcd(a_1, \ldots, a_n).$$

For example, see steps 4 and 9 of ROW. Large multipliers x_i lead to large intermediate expression growth. In the following section we examine algorithms for reducing the size of the multipliers.

3. The Greatest Common Divisor Algorithm

The following material is included in Howell [1976]. We compute the gcd of n polynomials in pairwise fashion. That is, if $a_1, a_2, \ldots, a_n$ are polynomials, we compute the gcd as follows:

$$\begin{aligned} g_1 &\leftarrow \gcd(a_1, a_2) \\ g_2 &\leftarrow \gcd(g_1, a_3) \\ &\vdots \\ g_{n-1} &\leftarrow \gcd(g_{n-2}, a_n) \end{aligned}$$

Here, g_{n-1} is $\gcd(a_1, a_2, \ldots, a_n)$. We can easily show that if we order the polynomials $a_1, \ldots, a_n$ so that the degree of a_1 is largest and the degree of a_n is smallest, then the bound on $\sum_{i=1}^{n} \deg(x_i)$, the sum of the degrees of the multipliers, is smaller than with the opposite ordering, that is, the smallest to largest ordering. Also the bound on $\max_i \deg(x_i)$ is smaller.

If, in addition to computing the g_i above, we save the multipliers, w_{i+1} and y_{i+1} at each step so that $g_i = \gcd(g_{i-1}, a_{i+1}) = w_{i+1} g_{i-1} + y_{i+1} a_{i+1}$ then we can compute the multipliers z_i so that $\gcd(a_1, \ldots, a_n) = z_1 a_1 + \ldots + z_n a_n$ as follows:

$$
\begin{aligned}
z_n &= y_n \qquad\qquad w'_n = w_n \\
&\left.\begin{aligned}
z_i &= y_i\, w'_{i+1} \\
w'_i &= w_i\, w'_{i+1}
\end{aligned}\right\} \quad i = n-1, \ldots, 2 \\
z_1 &= w'_2
\end{aligned}
$$

A smaller bound for the degrees of the multipliers is obtained when we modify the algorithm as follows:

$$
\begin{aligned}
z_n &= y_n - v_n \cdot \text{quotient}(y_n/v_n) \\
w'_n &= w_n - u_n \cdot \text{quotient}(y_n/v_n) \\
&\left.\begin{aligned}
z_i &= y_i\, w'_{i+1} - v_i \cdot \text{quotient}(z_i\, w'_{i+1}/v_i) \\
w'_i &= w_i\, w'_{i+1} - u_i \cdot \text{quotient}(z_i w'_{i+1}/v_i)
\end{aligned}\right\} \quad i = n-1, \ldots, 2
\end{aligned}
$$

where $u_i = -a_i/g_{i-1}$ and $v_i = g_{i-2}/g_{i-1}$

A related discussion is given in Bradley [1970].

These algorithms have been coded in ALTRAN.

5. References

Bradley, G. H., "Algorithm and Bound for the Greatest Common Divisor of n integers," Comm. ACM, V. 13, 1970, 433-436.

Bradley, G. H., "Algorithms for Hermite and Smith Normal Matrices and Linear Diophantine Equations," Math. Comp. V. 25, 897-907.

Gantmacher, F. R., "The Theory of Matrices," V. I, Chelsea, New York, 1960.

Howell, J. A., "An Algorithm for Computing the Greatest Common Divisor of n Polynomials," Los Alamos Scientific Laboratory report, LA-UR, 1976.

Turnbull, H. W. and A. C. Aitken, "An Introduction to the Theory of Canonical Matrices," Dover, New York, 1961.